一种通过最优子集法结合数据驱动的自适应权重处理小样本超高维类不平衡问题的分类方法

这是测试投稿文件，盗版必究

目录

1 摘要

随着科学技术的不断发展，计算机性能的不断提升，现如今的数据呈现指数级增长。如何很好地处理这些海量的数据, 以及通过数据处理方法和统计方法，从这些海量的数据之中获取有用的信息，已经成为了现在统计学家们关心的问题。在海量数据的情况下，常常会出现变量数远大于样本数的情况 $p >> n$，这里 p 为变量个数，也就是维数，n 为样本数。这是十分棘手的，因为在 $p >> n$ 的情况下，很多传统统计中的性质就失去了意义，所建立的统计模型也无法很好的解释事实现象。因此，需要对数据进行变量选择，这里的变量选择是统计中的 Variale selection，随着深度学习地不断发展，现在变量选择更多地情况下被成为特征筛选即 Feature selection。。但从个人角度认为，这两者实际上是一样的，都是从海量的特征之中通过处理方法找到真正起到影响作用的变量或者叫特征。随着变量选择方法的不断提升和改进，我们面对超高维数据的时候也可以进行处理，并且还可以获得一些比较好的性质，如oracle性质等。为此，本文尝试使用统计学界的变量选择方法与机器学习方法结合来处理超高维情况下的一些问题。本篇文章结构如下，第二章对前人的方法进行总结叙述，第三章介绍了最优子集方法，第四章通过实际数据进行方法的检验，最终进行了总结和讨论。

2 文献综述

关于变量选择这一问题，从统计学界的角度来看，比较传统的有逐步回归法，最优子集法等，这些方法可以处理比如多重共线性等问题，但是在 p 很大的时候就不是那么有效了, 为了处理这一棘手的问题 Tibshirani[1] 在 1996 年提出了lasso方法，在回归中加入 L_1 惩罚项，巧妙地实现了变量选择。这一方法开拓了变量选择这一方向，在以后的很长一段时间中，许多学者依然是从加入惩罚项这一角度入手，来解决变量选择问题。lasso方法从一定程度上解决了高维情况下变量选择的问题，但是由于 L_1 范数自身的特性，导致其估计出来的参数无法避免有偏性。为此 Fan 和 Li[2] 在 2001 年提出 SCAD 方法，并且提出oracle性质。假设 $\mathcal{A} = \{j : \beta_i^* \neq 0\}$ 并假设 $|\mathcal{A}| = p_0 < p$, 因此假设真实的模型是取决于自变量的子集，即这里 $\mathcal{A}$ 为真实的模型。这里将 $\widehat{\beta}(\delta)$ 记为通过方法 δ 拟合出来的参数估计值。$\widehat{\beta}(\delta)$ 的oracle性质即为：

- 能够找到真实的子集，$\lim_n P\left(\mathcal{A}_n^* = \mathcal{A}\right) = 1$
- 有渐进正态性质，$\sqrt{n}\left(\beta(\delta)_{\mathcal{A}} - \beta_{\mathcal{A}}^*\right) \to_d \mathrm{N}\left(\mathbf{0}, \Sigma^*\right)$。依分布收敛与真实的参数收敛到正态分布。这里 Σ^* 是真实子模型的协方差矩阵。

在 Fan 提出oracle性质之后，传统的lasso方法并不能满足oracle性质，Fan 提出的 SCAD 方法能够满足oracle性质，从而进一步提升了变量选择方法的性能。在以后的很长一段时间，一个变量选择方法是否能够满足oracle性质成为了判别其好坏的主要评判标准。Zou[3] 等人在 2005 年提出弹性网模型，创新地将lasso和ridge方法的 L_1 和 L_2 范数结合到一起，弹性网模型能够较好的解决数据中存在较强相关的两个变量的情况，Zou 通过数学方法巧妙地证明了弹性网模型能够很好的解决这一问题，并在处理医学基因数据中得到了很好的结果。Efron 等人[4] 在 2005 年提出了least angel regression方法，该算法提升了lasso方法，并且可以快速计算出lasso方法的参数估计值，并且解释了lasso和向前回归法的联系。Meinshausen[5] 等人在 2006 年提出了图lasso方法，进一步提升了lasso方法的使用范围。接着 Zou[6] 等人提出adaptive lasso方法，该方法通过创新地将惩罚项之前加入根据样本计算的系数 ω，这里 ω 可以是根据样本计算出的普通最小二乘的系数，即 $\arg\min_{\beta} \left\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\right\|^2 + \lambda \sum_{j=1}^{p} w_j \left|\beta_j\right|$，通过进行加权的方法，使传统的lasso方法有了适应性，并且 Zou 证明了adaptive lasso方法具有oracle性质，从而极大的提升了lasso方法的性能。Candes[7] 等人在 2007 年提出dantzig selector方法，该方法即是求解 $\min \|\hat{\beta}\|_1 \quad \text{s.t.} \left\|X'r\right\|_\infty \leq \lambda_p \cdot \sigma$, 该方法限制目标函数的梯度在 λ 内，该方法与lasso方法差别非常小，可以看作是受lasso的 kkt 条件启发而给出的优化范式。

至此为止，本文讨论了从 Tibshirani 开始开创的使用惩罚项的变量选择方法，从 1996 年开始，这一方法基本占据了变量选择的主流，并且在实际应用当中取得了很好的效果。并且变量选择的方法也与兴起的机器学习产生了交叉，通过使用统计中的变量选择方法来处理机器学习中的问题也成为了一种方式。本文对传统的变量选择方法总结如下，只列出部分方法。

在 2008 年，Fan[8] 提出了 Sure independence screening 方法（以下简称 SIS 方法），该方法将变量选择从传统的惩罚项的改进方面提供了一个新的视野，其实 SIS 方法就是对每个变量计算其与因变量 y 的皮尔逊相关系数 r, 然后根据相关系数的大小排序进行变量的选择。这种看起来十分简单粗暴的办法，在处理超高维数据时也会有不错的效果。Fan 在其中提出了

表 1: 表一传统变量选择方法比较

传统变量选择比较			
方法	无偏性	连续性	稀疏性
LASSO	无	无	有
SCAD	有	有	有
Ridge	有	有	无
硬门限	有	无	有

SIS 性质如下，将 $\mathcal{M}_*$ 记为真实的模型，$\mathcal{M}_\gamma$ 记为最终的模型。

$$P\left(\mathcal{M}_* \subset \mathcal{M}_\gamma\right) \to 1 \quad \text{as } n \to \infty$$

即在样本量 n 趋近于无穷的时候，真实模型属于最终模型的概率趋近于 1。这种性质能够很好地保证最终筛选出来的模型与真实的模型有一致性。在 Fan 提出该方法之后其本人在 2010 年[9] 提出了对 SIS 方法的改进，将 SIS 方法拓展到广义线性模型之中，并改变了 SIS 方法中的参数估计方法，从之前的相关系数拓展到极大边际似然统计量，进一步提升了 SIS 方法的引用范围。之后 Li[10] 等人在 2012 年提出了基于距离关联的 SIS 方法（简称为 DC-SIS)，将 Fan 之前的单纯使用皮尔逊相关系数的方法拓展到了更大的范围，该方法放宽了 SIS 方法对自变量和因变量的限制条件，但是 DC-SIS 方法失去了 SIS 性质，也有一定的局限性。

通过对前人方法的总结，传统的方法在处理变量选择问题的时候基本都不可避免的会受到模型假定条件的限制或者统计性质的局限，其实在统计方法内，对变量选择最好的方法是最优子集法，但是其高昂的计算成本让其在之前成为了一种相对不可行的方法，但是 Zhu[11] 在 2020 年提出了一种计算最优子集的算法，将其从不可能变为了可能。

3 方法简介

3.1 最优子集法简介

最优子集法即`best subset selection`, 最早可见 Hocking[12] 等人 1967 年的文章，其思想十分简单。从零号模型（null model）M_0 开始，这

个模型只有截距项而没有任何自变量。然后用不同的特征组合进行拟合，从特征中分别挑选出一个最好的模型（RSS 最小或 R^2 最大），也就是包含 1 个特征的模型 M_1，包含 2 个特征的模型 M_2，直至包含 p 个特征的模型 M_p。然后从这总共 p+1 个模型中选出其中最好的模型（根据交叉验证误差，C_p，BIC 或 adjusted R^2）（注：为什么不能用 RSS 或 R^2 来衡量？因为增加任何特征，模型的训练 RSS 只会变小，R^2 只会增大）。这个最好模型所配置的特征就是筛选出的特征。最优子集法在理想的条件下可以筛选出最好的特征集合出来，但是其高昂的计算成本是阻碍其应用的主要问题。Zhu[11] 等人在 2021 年提出了一种最优子集法的多项式算法，证明了在一定条件下，该算法具有以下三个优良性质:

- 计算复杂度是多项式的
- 选择出来的子集能够覆盖真实的集合
- 该算法的解是全局最优的

作者将该方法称为`Adaptive Best-Subset Selection`以下简称`BeSS`, 在作者提出的 SIC(special information criterion) 准则下，该方法的模型选择连续性得到了证明。SIC 准则如下:

$$\mathrm{SIC}(\mathcal{A}) = n \log \mathcal{L}_{\mathcal{A}} + |\mathcal{A}| \log(p) \log\log n$$

其中 $\mathcal{A}$ 为筛选出来的特征的集合。

3.2 剪接方法简介

首先定义一些变量名称，$\beta = \left(\beta_1, \ldots, \beta_p\right)^\top \in \mathbb{R}^p$，将 ℓ_q 定义为：$\|\beta\|_q = \left(\sum_{j=1}^p \left|\beta_j\right|^q\right)^{1/q}$ $q \in [1, \infty)$。$\mathcal{S} = \{1, \ldots, p\}$，对任何 $\mathcal{A} \subseteq \mathcal{S}$, 记 $\mathcal{A}^c = \mathcal{S} \backslash \mathcal{A}$ 作为 $\mathcal{A}$ 的补集,$|\mathcal{A}|$ 作为他的基。将 β 的子集定义为 $\mathrm{supp}(\beta) = \left\{j : \beta_j \neq 0\right\}$。对一个指标集 $\mathcal{A} \subseteq \{1, \ldots, p\}$，$\beta_{\mathcal{A}} = \left(\beta_j, j \in \mathcal{A}\right) \in \mathbb{R}^{|\mathcal{A}|}$。对一个矩阵 $X \in \mathbb{R}^{n \times p}$ 定义 $X_{\mathcal{A}} = (X_j, j \in \mathcal{A}) \in \mathbb{R}^{n \times |\mathcal{A}|}$。对任何一个向量 t 和任何一个集合 $\mathcal{A}$,$t^{\mathcal{A}}$ 是一个第 j 个元素 $\left(t^{\mathcal{A}}\right)_j$ 如果 $j \in \mathcal{A}$ 就等于 t_j，否则为 0。

在文章中，作者最重要的贡献是提出了剪接方法，即`splicing method`，通过使用该种方法，显著的提升了最优子集算法的性能，使其在当前的环境

下变的可行。考虑 ℓ_0 限制最小化问题：

$$\min_{\beta} \mathcal{L}_n(\beta), \quad \text{s. } t\|\beta\|_0 \leq s$$

其中：$\mathcal{L}_n(\beta) = \frac{1}{2n}\|y - X\beta\|_2^2$。在不考虑全局损失的情况下，我们考虑 $\|\beta\|_0 = s$。给定一个初始集合 $\mathcal{A} \subset \mathcal{S} = \{1, 2, \ldots, p\}$，且 $|\mathcal{A}| = s$，记 $\mathcal{I} = \mathcal{A}^c$ 并且计算

$$\hat{\beta} = \arg\min_{\beta_{\mathcal{I}}=0} \mathcal{L}_n(\beta)$$

将 $\mathcal{A}$ 和 $\mathcal{I}$ 定义为激活集合和非激活集合，这里的激活就是在真实集合中的意思。在给定了 $\mathcal{A}$ 和 $\hat{\beta}$ 之后，可以定义两种损失如下：

- 后退损失

$$\xi_j = \mathcal{L}_n\left(\hat{\beta}^{\mathcal{A}\backslash\{j\}}\right) - \mathcal{L}_n\left(\hat{\beta}^{\mathcal{A}}\right) = \frac{X_j^\top X_j}{2n}\left(\hat{\beta}_j\right)^2$$

- 前进损失：

$$\zeta_j = \mathcal{L}_n\left(\hat{\beta}^{\mathcal{A}}\right) - \mathcal{L}_n\left(\hat{\beta}^{\mathcal{A}} + \hat{t}^{\{j\}}\right) = \frac{X_j^\top X_j}{2n}\left(\frac{\hat{d}_j}{X_j^\top X_j/n}\right)^2$$

其中：$\hat{t} = \arg\min_t \mathcal{L}_n\left(\hat{\beta}^{\mathcal{A}} + t^{\{j\}}\right), \hat{d}_j = X_j^\top(y - X\hat{\beta})/n$

直观的而言，对 $j \in \mathcal{A}$ 一个大的 ξ_j 说明这个变量是潜在重要的。但是由于子集大小不同，这两个损失是无法比较的。然而如果将 $\mathcal{A}$ 中的一些不是那么相关的变量和 $\mathcal{I}$ 中一些重要的变量交换，这也许能够获得比较好的结果，这就是剪接法的思想所在。特别的，对任何给定的 $k <= s$ 定义如下：

$$\mathcal{A}_k = \left\{ j \in \mathcal{A} : \sum_{i \in \mathcal{A}} \mathrm{I}\left(\xi_j \geq \xi_i\right) \leq k \right\}$$

$$\mathcal{I}_k = \left\{ j \in \mathcal{I} : \sum_{i \in \mathcal{I}} \mathrm{I}\left(\zeta_j \leq \zeta_i\right) \leq k \right\}$$

通过交换 $\mathcal{A}_k$ 和 $\mathcal{I}_k$, 从而实现对 $\mathcal{A}$ 和 $\mathcal{I}$ 的切片，得到了新的集合：

$$\tilde{\mathcal{A}} = (\mathcal{A}\backslash\mathcal{A}_k) \cup \mathcal{I}_k$$

记 $\tilde{\mathcal{I}} = \tilde{\mathcal{A}}^c, \tilde{\beta} = \arg\min_{\beta_{\tilde{\mathcal{I}}}=0} \mathcal{L}_n(\beta)$，并且 $\tau_s > 0$ 为阈值。如果 $\tau_s < \mathcal{L}_n(\hat{\beta}) - \mathcal{L}_n(\tilde{\beta})$ 则说明 $\tilde{\mathcal{A}}$ 是优于 $\mathcal{A}$。通过这样的方法则可以更新集合 $\mathcal{A}$ 指导损失函

数不能够通过剪接方法来进行提升。还有的问题就是设定初始集。通常来说我们选定第一批 s 个特征，这些特征是与 y 关联程度最大的特征。设 k_{max} 为剪接最大尺寸，$k_{max} < s$, 接下来的算法演示了如何计算具体的方法。

Algorithm 1: 计算初始集合算法

Input: 输入参数 X,y, 正整数 k_{max} 和阈值

$$\tau_s,\mathcal{A}^0 = \left\{ j : \sum_{i=1}^{p} \mathrm{I}\left(\left| \frac{x_j^\top y}{\sqrt{x_j^\top x_j}} \right| \leq | \frac{x_i^\top y}{\sqrt{x_i^\top x_i}} \leq \mathrm{s} \right\} , \mathcal{I}^0 = \left(\mathcal{A}^0\right)^c ,;$$

$\left(\beta^0, d^0\right) : \beta^0_{\mathcal{I}^0} = 0,\ d^0_{\mathcal{A}^0} = 0,\ \beta^0_{\mathcal{A}^0} = \left(X^\top_{\mathcal{A}^0} X_{\mathcal{A}^0}\right)^{-1} X^\top_{\mathcal{A}^0} y_I$

$d^0_{\mathcal{I}^0} = X^\top_{\mathcal{I}^0}\left(y - X\beta^0\right)/n$

Output: $(\hat{\beta}, \hat{d}, \hat{\mathcal{A}}, \hat{\mathcal{I}}) = \left(\beta^{m+1}, d^{m+1}\mathcal{A}^{m+1}, \mathcal{I}^{m+1}\right)$

计算初始集合;

for $m = 0, 1, 2, 3....$ **do**

$\left(\beta^{m+1}, d^{m+1}, \mathcal{A}^{m+1}, \mathcal{I}^{m+1}\right) = \mathrm{Splicing}\left(\beta^m, d^m, \mathcal{A}^m, \mathcal{I}^m, k_{\max}, \tau_s\right);$

end

if $(\mathcal{A}^{m+1}, \mathcal{I}^{m+1}) = (\mathcal{A}^m, \mathcal{I}^m)$ **then**

stop;

end

Algorithm 2: 剪接算法

Input: $\beta, d, \mathcal{A}, \mathcal{I}, k_{\max}, \tau_s$， $L_0 = L = \frac{1}{2n}\|y - X\beta\|_{2'}^2$，
$\xi_j = \frac{x_j^\top x_j}{2n}\left(\beta_j\right)^2, \zeta_j = \frac{x_j^\top x_j}{2n}\left(\frac{d_j}{x^\top x_i/n}\right)^2, j = 1, \ldots, p$

Output: $(\hat{\beta}, \hat{d}, \hat{\mathcal{A}}, \hat{\mathcal{I}})$

进行剪接算法;

for $k = 0, 1, 2, 3....k_{max}$ **do**

$\mathcal{A}_k = \left\{j \in \mathcal{A} : \sum_{i \in \mathcal{A}} \mid \left(\xi_j \geq \xi_i\right) \leq k\right\}$;

$\mathcal{I}_k = \left\{j \in \mathcal{I} : \sum_{i < \mathcal{I}} \mid \left(\zeta_j \leq \zeta_i\right) \leq k\right\}$;

让 $\tilde{\mathcal{A}}_k = (\mathcal{A} \backslash \mathcal{A}_k) \cup \mathcal{I}_k, \tilde{\mathcal{I}}_k = (\mathcal{I} \backslash \mathcal{I}_k) \cup \mathcal{A}_k$ 并求;

$\tilde{\beta}_{\tilde{\mathcal{A}}_k} = \left(x_{\tilde{\mathcal{A}}_k}^\top x_{\tilde{\mathcal{A}}_k}\right)^{-1} X_{\tilde{\mathcal{A}}_k}^\top y_1, \tilde{\beta}_{\tilde{\mathcal{I}}_k} = 0$;

$\tilde{d} = X^\top (y - X\tilde{\beta})/n, \quad \mathcal{L}_n(\tilde{\beta}) = \frac{1}{2n}\|y - X\tilde{\beta}\|_2^2$;

end

if $L > \mathcal{L}_n(\tilde{\beta})$ **then**

$(\hat{\beta}, \hat{d}, \hat{\mathcal{A}}, \hat{\mathcal{I}}) = \left(\tilde{\beta}, \tilde{d}, \tilde{\mathcal{A}}_k, \tilde{\mathcal{I}}_k\right)$;

$L = \mathcal{L}_n(\tilde{\beta})$.;

end for

end

if $L_0 - L < \tau_s$ **then**

$(\hat{\beta}, \hat{d}, \hat{\mathcal{A}}, \hat{\mathcal{I}}) = (\beta, d, \mathcal{A}, \mathcal{I})$

end

输出结果

通过如上两个算法，Zhu 等人巧妙地将最优子集法的计算问题转化为了可以计算的实际应用算法。

3.3 类不平衡问题加权方法

在进行分类问题时，经常会遇到类不平衡问题，具体表现为某一类的个数明显的超过了另一类，尤其是在处理二分类问题时。对类不平衡问题，常见地处理方法有加权方法和抽样方法。抽样方法有欠采样和过采样，对少的类进行过采样，对多的类进行欠采样，通过这样的方法来使类不平衡的问题得到处理，详见 Li[13],Tang[14] 和 Zou[15] 等人这些前人研究了类不平衡问题在支持向量机中的解决方法，这里我们受到他们思想的启发。

另一种处理类不平衡问题的方法是加权方法，详见 Hwang[16] 等人。将正例点个数记为 N_{pos}，将负例点个数记为 N_{neg}, 最常用的权重为如下权重:

$$w_i = \begin{cases} 1/N_{\text{pos}} & \text{if } y_i = 1 \\ 1/N_{\text{neg}} & \text{otherwise} \end{cases}$$

然而，Hwang 设计了一种不同的权重方式，来提升算法的收敛速度，新的权重为：

$$w_i = \begin{cases} 1 & \text{if } y_i = 1, \quad N_{pos} \geq N_{neg} \\ N_{neg}/N_{pos} & \text{if } y_i = 1, \quad N_{pos} < N_{neg} \\ N_{pos}/N_{neg} & \text{if } y_i = -1, \quad N_{pos} \geq N_{neg} \\ 1 & \text{if } y_i = -1, \quad N_{pos} < N_{neg} \end{cases}$$

该权重不仅能够保留比率，并且能够使收敛速度更快。本文采用该思想，对数据进行加权处理。

4 模拟实证

本文考虑使用 BeSS 方法和加权方法来处理超高维且类不平衡条件下的二分类问题，在该条件下，将 BeSS 方法的回归模型设置为 logstic 模型，从而得到分类结果。首先使用较为标准的数据集合进行模拟计算，然后使用真实的数据集来验证我们的方法是否具有可行性。在这里我们将我们的方法命名为`WBESS(weighted bess)` 使用 UCI 数据库的 RNA 序列数据，共有 800 个样本，20000 个特征数, 每个样本的类别是其所患肿瘤的类型不同，将该数据集合使用不同的方法进行建模，得到二分类结果，从而验证 BeSS 方法和加权方法对该类问题是否能够有所提升。

4.1 模拟计算

在这个例子中，本文生成 250 个正例和 350 个负例点，正例点的均值为 +2，负例点的均值为-2, 数据维度为 400 维。使用自适应加权支持向量机得到的混淆矩阵结果如下。横轴为预测结果，纵轴为原始标签。总共有三种情况如下所示:

- 情况 1：正例 250 个，负例 350 个，维度为 400

- 情况 2：正例 250 个，负例 500 个，维度为 400
- 情况 3：正例 250 个，负例 500 个，维度为 1000

4.1.1 情况 1

表 2: 使用自适应加权支持向量机分类结果

	-1	1
-1	57	14
1	13	36

使用传统支持向量机结果如下表所示。

表 3: 使用传统支持向量机混淆矩阵

	-1	1
-1	60	45
1	3	12

使用 logstic 回归结果如下：

表 4: 使用 logstic 回归结果

	-1	1
-1	47	17
1	19	37

使用 BeSS 方法结果的混淆矩阵如下：

表 5: 使用 BeSS 结果

	-1	1
-1	61	8
1	5	46

计算各个方法的指数如下表所示：

表 6: 250 个正例和 300 个负例情况下各种方法参数对比

method	Sensitivity	Specificity	FPR	FNR	ACC
awensvm	73.46	80.28	18.30	18.57	77.50
svm	80.00	57.14	2.80	4.76	60.00
BeSS	90.19	88.40	7.20	7.57	89.16
logstic	66.07	73.43	29.68	28.78	70.00

4.1.2 情况 2

接着本文生成 250 个正例点和 500 个负例点来对比各种方法的效果，结果如下表所示。

表 7: 250 个正例和 500 负例情况下各种方法指标比对

method	Sensitivity	Specificity	FPR	FNR	ACC
awensvm	69.49	93.40	6.59	30.50	84.00
svm	12.50	100.00	0.00	42.60	72.00
logstic	89.58	76.47	23.52	10.41	78.28
BeSS	100.00	100.00	0.00	0.00	100.00

可视化结果如下图所示。

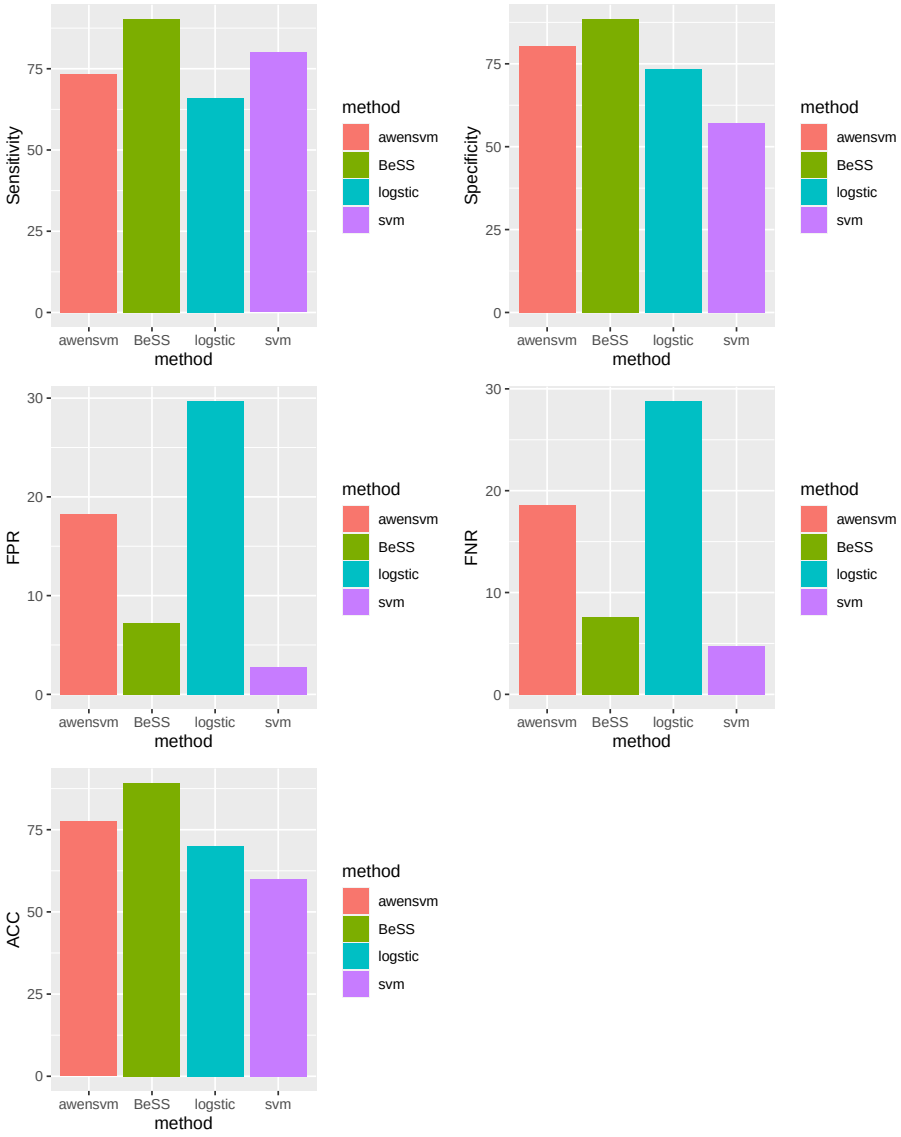

图 1: 250 个正例和 300 个负例情况下各种方法比较图

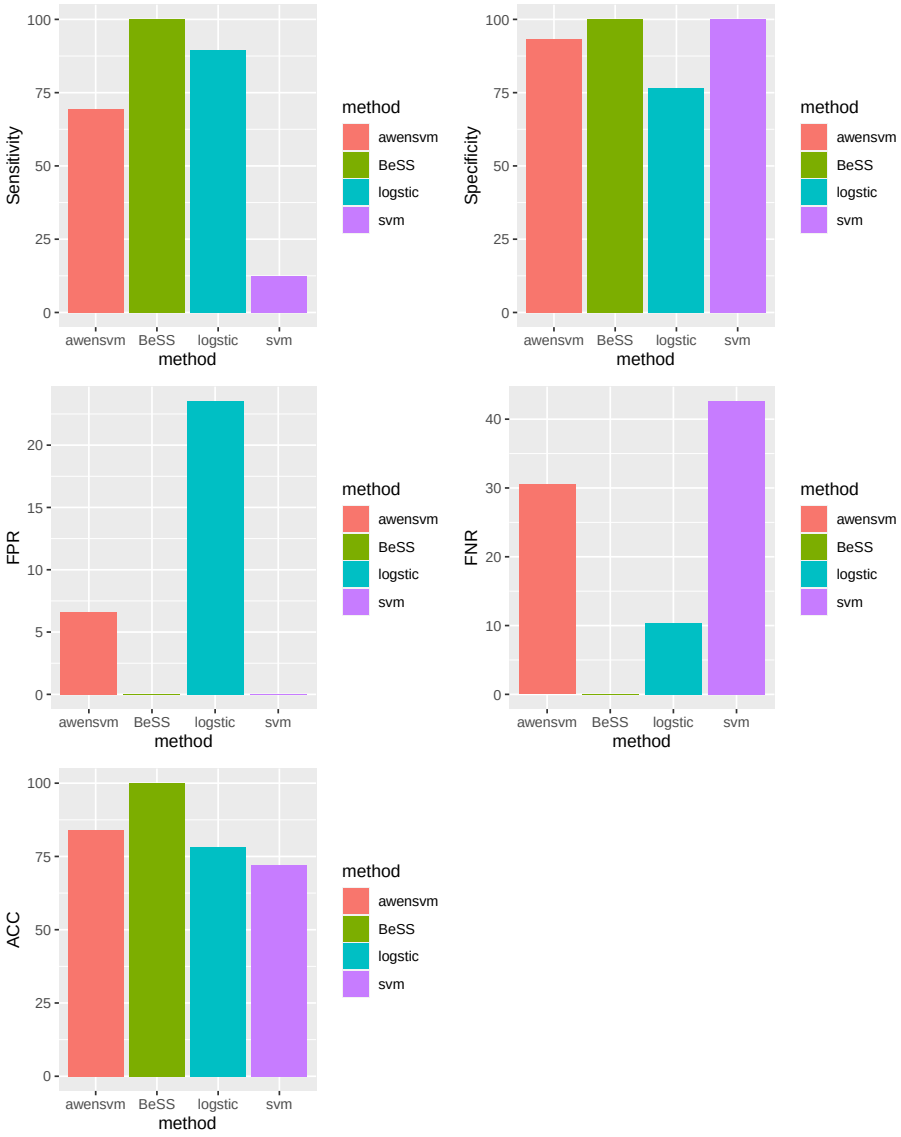

图 2: 250 个正例和 500 个负例情况下各种方法比较图

4.1.3 情况 3

接着本文将正例点和负例点控制在 250 和 500 个，将数据的维数提升到 1000 维，来比较各种方法的好坏。

表 8: 250 个正例和 500 负例在 1000 维情况下各种方法指标比对

method	Sensitivity	Specificity	FPR	FNR	ACC
awensvm	67.24	91.30	8.69	32.75	82.00
svm	0.00	1.00	0.00	1.00	67.33
logstic	61.22	48.51	51.48	38.77	52.66
BeSS	92.45	1.00	0.00	7.54	97.33

可视化结果如下图所示。

4.1.4 总结

通过在三种不同情况下对各种方法进行对比如下表所示。

表 9: 各种方法在不同情况下各项指标对比

方法	awensvm			SVM			logstic			BeSS		
指标	condition1	condition2	condition3	condition1	condition2	condition3	condition1	condition2	condition3	condition1	condition2	condition3
Sensitivity	73.46	69.49	67.24	80	12.5	0	66.07	89.58	61.22	90.19	100	92.45
Specificity	73.46	93.4	91.3	57.14	100	1	73.43	76.47	48.51	88.4	100	1
FPR	18.3	6.59	8.69	2.8	0	0	29.68	23.52	51.48	7.2	0	0
FNR	18.57	30.5	32.75	4.76	42.6	1	28.78	10.41	38.77	7.57	0	7.54
ACC	77.5	84	82	60	72	67.33	70	78.28	52.66	89.16	100	97.33

通过对不同方法在三种情况下各个指标的对比，发现传统的分类方法如 SVM 和 logstic 方法，在比较正常的情况下还是有一定的效果。但是当类不平衡情况加重的情况下和维度扩张到 $p >> n$ 的情况下时，传统的分类方法的各项指标会有显著的显著下降，造成了其无法正常的进行分类。而使用自适应加权弹性网支持向量机和BeSS方法可以克服这些情况，在不同情况下的各项指标显示，BeSS方法都显著优于AWENASVM方法。通过模拟计算，本文证明了BeSS方法在类不平衡和超高维情况下的优异表现。

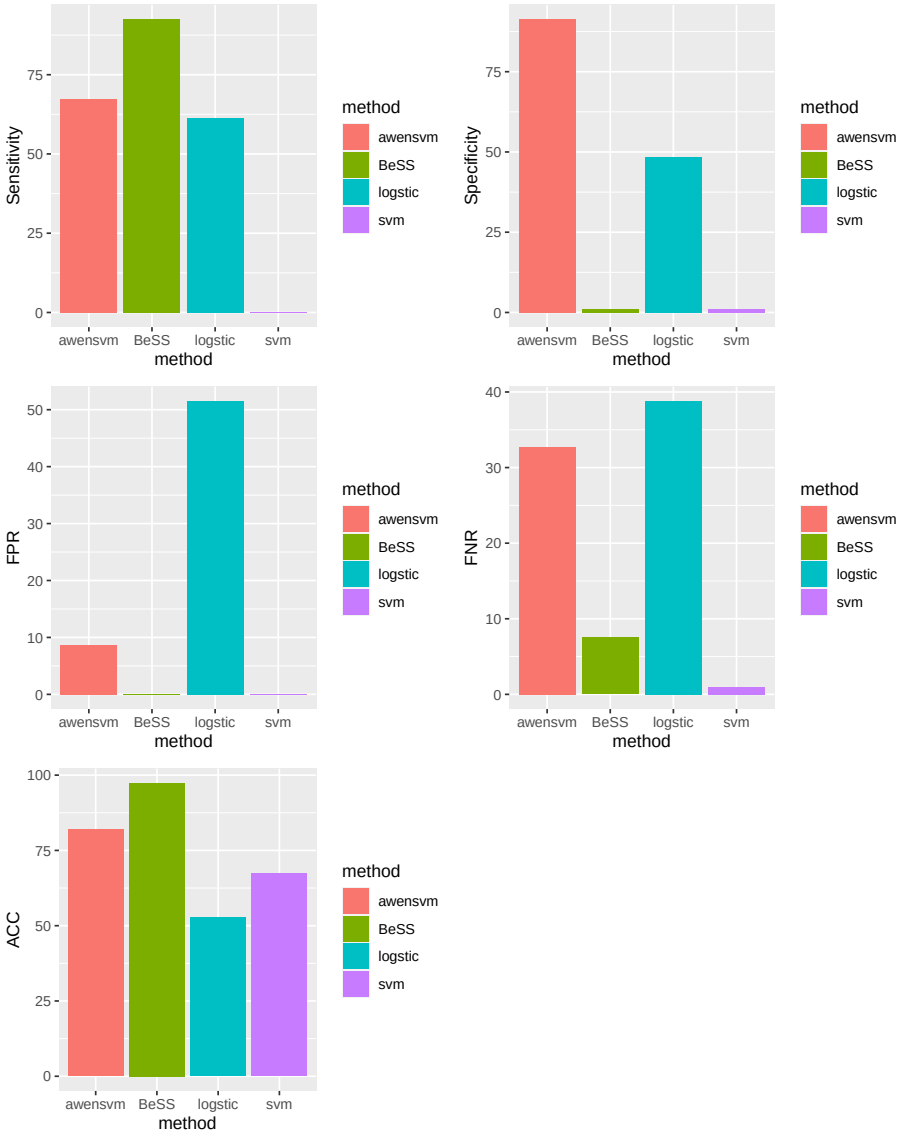

图 3: 250 个正例和 500 个负例在 1000 维情况下各种方法比较图

4.2 对实证数据进行未加权建模

本文首先对该数据进行原始 BeSS 方法建模，具体参数及内容详见 Wen[17], 本文中将回归模型设置为 “binomial”, 不进行权重加权时，每个样本默认的权重为 1。建模的结果如图 1 所示。

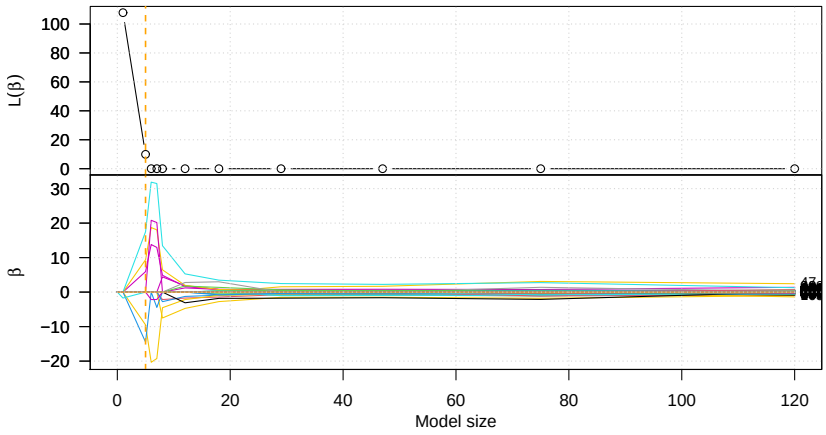

图 4: 没有进行加权建模模型过程图

从图 1 可以看出，随着模型大小的增加，$L(\beta)$ 会出现一个降低的趋势, 并且随着模型大小的变化，各个变量的系数也在图中有展示。该图横轴为模型规模即变量个数，上面第一幅图为损失函数的变化，下面表 2 为随着模型大小，各个特征系数。根据 Zhu[11] 等人提出的 SIC 准则，在最终选择模型的时候并不只看损失函数的大小。

可以看出，随着模型规模即 Df 的变化，各项模型参数也随之变化，在各项指标之中，综合考虑，从而找到效果最好的模型。通过 BeSS 方法最终建模结果如下

$$y = -1980.53 + 33.44 * x_{1317} + 45.08 * x_{1750} + 48.55 * x_{1885} - 48.88 * x_{2318} + 77.17 * x_{4092} - 3.11 * x_{1317}$$

脚标代表着筛选出的基因名称。因为是使用 logstic 回归模型, 将阈值设定为 0, 输出混淆矩阵，通过混淆矩阵来评判该模型的好坏。

表 10: 建模过程中各项指标比较

Df	Dev	AIC	BIC	EBIC
1	107.8423059	109.8423059	114.5281669	133.9361659
120	0.000503217	240.0005032	802.3038169	3131.263695
75	0.000234146	150.0002341	501.4398052	1957.039729
47	0.00052864	94.00052864	314.2359932	1226.411945
29	0.000561639	58.00056164	193.8905291	756.7224996
18	0.000278811	36.00027881	120.3457759	469.6897575
12	0.000197037	24.00019704	80.2305284	313.1265162
8	0.000555819	16.00055582	53.4874434	208.7514352
5	9.908629066	19.90862907	43.3379338	140.3779287
7	0.008687521	14.00868752	46.80971415	182.665707
6	0.008531852	12.00853185	40.12369753	156.5716914

通过混淆矩阵看出，在没有使用加权方法进行建模时，混淆矩阵中实际为 1 的类被误分的比较多，其中 $N_{TP} = 212, N_{FP} = 88, N_{TN} = 391, N_{FN} = 110$, 可以看出由于 $+1$ 例较少，造成了比较严重的误分类情况，该情况下模型的 ACC 计算如下：$P_{acc} = \frac{N_{TP}+N_{TN}}{N_{TP}+N_{TN}+N_{FP}+N_{FN}} = 0.75$

4.3 对实证数据进行加权分类建模

通过之前提到的加权方法对输出进行加权之后建模。具体流程基本如上。使用加权处理之后，模型输出的模型选择图如图 2 所示。从图二看出，使用加权方法进行数据处理之后，数据的损失函数相比没有加权之前有了更大的降低，因此本文认为使用加权方法处理高维类不平衡的分类问题是有一定效果的。

使用加权方法建立的最终模型如下所示:

$$y = -152.76 - 21.892 * x_{2951} - 26.25 * x_{6611} + 18.97 * x_{6876} - 12.21 * x_{8349} + 29.11 * x_{8665} + 7.36 * x_{16245}$$

通过使用加权的方法，筛选出的变量与没有加权的方法有了变化，这是因为每个样本的权重与没有加权的不一样了，这里的权重可以认为是数据的重要程度，在本文中，相当于是对较少的样本进行了加重处理，从而来针对类不平衡问题。接着看使用加权方法之后的混淆矩阵。使用加权处理之后，

表 11: 未进行加权建模的混淆矩阵

		预测	
		-1	1
实际	1	88	212
	-1	391	110

$N_{TP} = 242, N_{FP} = 58, N_{TN} = 429, N_{FN} = 72$，这时的模型的 $P_{acc} = 0.83$。通过对 ACC 值的比较发现，进行过加权处理的模型的 ACC 值提升了 8 个百分点，因此我们认为使用加权的方法能够对小样本下超高维数据类不平衡情况下提升分类准确率有一定的作用。

表 12: 加权处理后的混淆矩阵

		预测	
		-1	1
实际	1	58	242
	-1	429	72

5 总结

本文通过将最新的最优子集法和数据驱动的自适应加权方法进行结合，讨论了该种方法的可行性，并在实际数据中通过模拟实验证明了该思想确

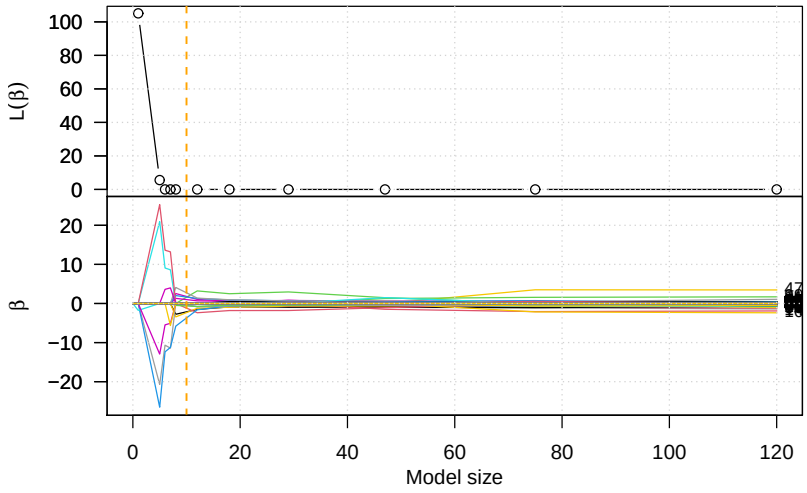

图 5: 进行加权建模模型过程图

实有一定的提升。但是由于种种原因限制，小样本和超高维数据不仅是有分类问题，也会有回归问题和聚类等问题，这些问题本文没有考虑到。结合现在实际情况，该种方法目前在医学诊断上面有不错的表现，但是本文认为，该种方法应该可以与目前比较火热的深度学习进行融合，如何在训练数据有限的情况下尽可能的建立最好的模型，来达到最好的效果，这个问题应该是比较有意义的。另外，关于最优子集算法这一问题，可以详细参考 Fan[18] 等人的文章，在文章中详细证明了关于最优子集法的相关问题，能够对深刻理解最优子集算法有很好的提升。

最后，本文在撰写过程中不免会有许多疏漏和考虑不周的情况，还请各位读者指正。

参考文献

[1] TIBSHIRANI R. Regression Shrinkage and Selection Via the Lasso[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1996, 58(1): 267–288.

[2] FAN J, LI R. Variable selection via nonconcave penalized likelihood and its oracle properties[J]. Journal of the American Statistical Association, 2001, 96(456): 1348–1360.

[3] ZOU H, HASTIE T. Erratum: Regularization and variable selection via the elastic net (Journal of the Royal Statistical Society. Series B: Statistical Methodology (2005) 67 (301-320))[J]. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 2005, 67(5): 768.

[4] EFRON B, HASTIE T, JOHNSTONE I, 等. Least angle regression[J]. Annals of Statistics, 2004, 32(2): 407–499.

[5] MEINSHAUSEN N, BÜHLMANN P. High-dimensional graphs and variable selection with the Lasso[J]. Annals of Statistics, 2006, 34(3): 1436–1462.

[6] ZOU H. The adaptive lasso and its oracle properties[J]. Journal of the American Statistical Association, 2006, 101(476): 1418–1429.

[7] CANDES E, TAO T. The Dantzig selector: Statistical estimation when p is much larger than n[J]. Annals of Statistics, 2007, 35(6): 2313–2351.

[8] FAN J, LV J. Sure independence screening for ultrahigh dimensional feature space[J]. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 2008, 70(5): 849–911.

[9] FAN J, SONG R. Sure independence screening in generalized linear models with NP-dimensionality[J]. Annals of Statistics, 2010, 38(6): 3567–3604.

[10] LI R, ZHONG W, ZHU L. Feature screening via distance correlation learning[J]. Journal of the American Statistical Association, 2012, 107(499): 1129–1139.

[11] ZHU J, WEN C, ZHU J, 等. A polynomial algorithm for best-subset selection problem[J]. Proceedings of the National Academy of Sciences of the United States of America, 2021, 117(52): 33117–33123.

[12] HOCKING R R, LESLIE R. Selection of the best subset in regression analysis[J]. Technometrics, Taylor & Francis Group, 1967, 9(4): 531–540.

[13] LI P, QIAO P-L, LIU Y-C. A hybrid re-sampling method for SVM learning from imbalanced data sets[C]//2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery. IEEE, 2008, 2: 65–69.

[14] TANG Y, ZHANG Y-Q, CHAWLA N V, 等. SVMs modeling for highly imbalanced classification[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), IEEE, 2008, 39(1): 281–288.

[15] ZOU S, HUANG Y, WANG Y, 等. SVM learning from imbalanced data by GA sampling for protein domain prediction[C]//2008 the 9th international conference for young computer scientists. IEEE, 2008: 982–987.

[16] HWANG J P, PARK S, KIM E. A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function[J]. Expert Systems with Applications, Elsevier, 2011, 38(7): 8580–8585.

[17] WEN C, ZHANG A, WANG X, 等. Bess: An R package for best subset selection in linear, logistic and cox proportional hazards models[J]. Journal of Statistical Software, 2020, 94(4): 1–24.

[18] FAN J, ZHU Z. When is best subset selection the《best》?[J]. arXiv, 2020.